



Sovellusten ohjelmointi ja käytettävyys

Oppimispäiväkirja

Antti Venetjoki

SISÄLLYS

1	Viikkoharjoitukset	3
1.1	Android -ympäristön asennus ja Hello World	3
1.1.1	Android -ympäristön asennus	3
1.1.2	HelloWorld- koodi	3
1.1.3	Todiste ohjelman ajosta.....	4
1.2	Jetpack Compose -tutustuminen.....	5
1.2.1	Koodi	5
1.2.2	Koodi ajettuna Android virtuaalikoneessa.....	8
1.3	Kotlin essentials – osa 1	9
1.3.1	Funktiot.....	9
1.3.2	Luokat ja metodit	9
1.3.3	Null safety.....	9
1.3.4	Extension funktiot	9
	Käytetyt lähteet	10

1 Viikkoharjoitukset

1.1 Android -ympäristön asennus ja Hello World

1.1.1 Android -ympäristön asennus

Asensin version Android Studiosta, joka sisälsi myös Android SDK ja tarvittavat työkalut. Asennusohjelma ohjasi automaattisesti SDK asennukseen. Huomasin, että prosessi asensi myös virtuaaliset Android-laitteet (AVD), joita käytetään sovellusten testaamiseen ilman fyysistä laitetta. Valitsin AVD Managerista laitteeksi Pixel 8 ja Android 12 -version, jossa on API-versio 31. Virtuaalikoneen käynnistämisessä huomasin, että laite voi olla hidas, mikä johtui osittain koneen resurssien rajoitteista.

1.1.2 HelloWorld- koodi

```
package com.example.helloworld
```

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.Text
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.runtime.Composable
```

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            HelloName()
        }
    }
}
```

```

    }
}

```

```
@Composable
```

```
fun HelloName() {
```

```
    Box(
```

```
        modifier = Modifier.fillMaxSize(), // Make the Box fill the screen
```

```
        contentAlignment = Alignment.Center // Center the content inside the Box
```

```
    ) {
```

```
        Text(text = "Antti Venetjoki")
```

```
    }
```

```
}
```

1.1.3 Todiste ohjelman ajosta



1.2 Jetpack Compose -tutustuminen

1.2.1 Koodi

```
package com.example.jetpackcomposestarter

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.foundation.border
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

val GLOBAL_FONT_SIZE_MAIN = 32.sp
val GLOBAL_ROUNDED_CORNER_RAD = 12.dp
val GLOBAL_SPACER_HEIGHT = 12.dp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            HelloApp()
        }
    }
}
```

```

}

@Preview(showBackground = true)
@Composable
fun HelloApp() {
    var textState by remember { mutableStateOf("Hello World") }
    Surface(
        modifier = Modifier
            .fillMaxSize(),
        color = Color.LightGray
    ) {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp),
            verticalArrangement = Arrangement.SpaceBetween,
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            HelloText("My App")
            Spacer(modifier = Modifier.height(GLOBAL_SPACER_HEIGHT))
            HelloText(textState, Modifier.weight(1f))
            Spacer(modifier = Modifier.height(GLOBAL_SPACER_HEIGHT))
            HelloButton(
                textState = textState,
                onTextChange = { newText -> textState = newText }
            )
        }
    }
}

```

```

@Composable
fun HelloText(str: String, columnModifier: Modifier = Modifier) {
    var customModifier = Modifier
        .fillMaxWidth()
        .border(

```

```

        width = 2.dp,
        color = Color.Black,
        shape = RoundedCornerShape(GLOBAL_ROUNDED_CORNER_RAD))
    .clip(RoundedCornerShape(GLOBAL_ROUNDED_CORNER_RAD))
    .background(Color.White)
    .padding(12.dp)

```

```

val combinedModifier = customModifier.then(columnModifier)

```

```

Column(
    modifier = combinedModifier,
    horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.Center
) {
    Text(
        text = str,
        fontSize = GLOBAL_FONT_SIZE_MAIN
    )
}

```

```

@Composable
fun HelloButton(
    textState: String,
    onTextChange: (String) -> Unit
) {
    Button(
        onClick = {
            val newText = if (textState == "Hello World") {
                "Hello User"
            } else {
                "Hello World"
            }
            onTextChange(newText)
        },

```

```

shape = RoundedCornerShape(GLOBAL_ROUNDED_CORNER_RAD),
colors = ButtonDefaults.buttonColors(
    containerColor = Color.Gray,
    contentColor = Color.Black
),
modifier = Modifier
    .fillMaxWidth()
    .border(
        width = 2.dp,
        color = Color.Black,
        shape = RoundedCornerShape(GLOBAL_ROUNDED_CORNER_RAD))
){
    Text(
        text = "Click To Say Hello",
        fontSize = GLOBAL_FONT_SIZE_MAIN
    )
}
}

```

1.2.2 Koodi ajettuna Android virtuaalikoneessa



1.3 Kotlin essentials – osa 1

1.3.1 Funktiot

```
fun add(num1: Int, num2: Int): Int {  
    return num1 + num2  
}
```

1.3.2 Luokat ja metodit

```
class Car(val brand: String, val year: Int) {  
    fun printCarInfo() {  
        println("Car brand: $brand, Year: $year")  
    }  
}
```

1.3.3 Null safety

```
fun main() {  
    val myString: String? = null  
  
    // Elvis-operaattori: Jos myString on null, käytetään oletusarvona "Default Value"  
    val result = myString ?: "Default Value"  
  
    println(result)  
}
```

1.3.4 Extension funktiot

```
fun String.addExclamation(): String {  
    return this + "!"  
}  
  
fun main() {  
    val greeting = "Hello Kotlin"  
    println(greeting.addExclamation())  
}
```

Käytetyt lähteet

<https://kotlinlang.org/docs/>